DOCUMENTATION : …/20

Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert. [sur 2 points]

Notre application est un jeu vidéo basé sur le principe de Space Invaders. Il faudra donc marquer le plus de points possibles en tuant les ennemis qui tombent. Pour cela, vous pouvez déplacer votre joueur, qui tir automatiquement, au moyen du gyroscope du téléphone, donc en l'inclinant pour aller à droite et à gauche afin de viser les ennemis. Vous disposez de trois vies, que vous perdez si un ennemi arrive à atteindre le bas de l'écran. Vous pourrez, sur la fenêtre d'accueil, consulter votre meilleur score pour essayer de le battre, mais également choisir la difficulté du jeu. Certains ennemis possèdent une seule vie, et d'autres deux, nous changeons donc la difficulté via le nombre d'ennemis de chaque type qui apparaît. Finalement, plus votre score augmente, plus le nombre d'ennemis à deux vies, donc plus difficile à tuer, augmente.

Je sais faire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application. [sur 5 points]

Cf notre diagramme de cas d'utilisation

Je sais concevoir un diagramme UML de qualité représentant mon application. [sur 7 points]

Cf notre diagramme de classe

Je sais décrire un diagramme UML en mettant en valeur et en justifiant les éléments essentiels. [sur 6 points]

Un élément essentiel de ce modèle est la classe TimerTask. Liée à la classe Game, elle va appeler de manière régulière sur l'UI Thread, la boucle de jeu, qui permet donc de faire fonctionner le jeu.

Nous pouvons également parler de l'organisation des entités. Chaque image apparaissant sur l'écran est de base un block, qui possède une image, une position etc.... A partir de cette classe mère on a pu définir des comportements spécifiques, comme le Player, qui représente le joueur, ou TaskBlock, qui englobe les ennemis et les tirs, qui se déplacent d'une certaine manière. Cela permet de faire du polymorphisme et donc gérer les mouvements de l'ensemble des blocks facilement avec la boucle de jeu.

Programmation : …/20

Je sais utiliser les Intent pour faire communiquer deux activités. [sur 1 point]

Nous utilisons par exemple les Intent pour démarrer l'activité de jeu, mais également lui passer la difficulté choisie par l'utilisateur dans le menu, pour pouvoir l'appliquer au jeu.
Preuve : lignes 29 de la « MainActivity ».

Je sais développer en utilisant le SDK le plus bas possible. [sur 1 point]

Oui, nous développons avec le SDK 15

Je sais distinguer mes ressources en utilisant les qualifier. [sur 1 point]

Nos ressources sont rangées dans le dossier qui lui correspond.
Exemple : res/drawable pour toutes nos images
Res/layout pour toutes nos fenêtres
Res/values pour nos valeurs statiques tel que les strings, les colors ou les styles

Je sais modifier le manifeste de l'application en fonction de mes besoins. [sur 1 point]

Oui, par exemple dans notre application nous souhaitons que la « MainActivity » se lance au démarrage de l'application. Dans le manifeste, nous avons donc déclaré cette activité comme étant celle à montrer au lancement de l'application. Nous avons aussi modifié le manifest pour changer d'icône et de thème.
Preuve : voir le manifest

Je sais faire des vues xml en utilisant layouts et composants adéquats. [sur 1 point]

Nous utilisons plusieurs vues xml utilisant par exemple des frameLayout, des radioGroup, TextView etc...
Preuves : les vues disponibles dans res→ layout

Je sais coder proprement mes activités, en m'assurant qu'elles ne font que relayer les évènements. [sur 1 point]

Dans nos activités, nous déléguons les modifications à d'autres classes, nos activités se contentent de récupérer les informations des vues, et également de modifier ses vues avec de nouveaux éléments.

Je sais coder une application en ayant un véritable métier. [sur 2 points]

Oui.
Preuve : la séparation de nos classes, activités etc dans des packages bien distincts.

Je sais parfaitement séparer vue et modèle. [sur 1 point]
La vue de jeu ne réalise aucune opération, elle se contente d'appeler le métier pour effectuer tous les calculs qui seront ensuite afficher sur l'écran

Je maîtrise le cycle de vie de mon application. [sur 1 point]
Lorsque l'application perd le focus, elle se met automatiquement en pause pour être repris par la suite (ex : recevoir un coup de téléphone)

Je sais utiliser le findViewById à bon escient. [sur 1 point]
Preuve : Pour écrire des informations dynamiquement dans une vue, nous utilisons findViewById pour récupérer une TextView et modifier à notre convenance, dans le code, son contenu.
Ligne 61 de GameActivityy

Je sais gérer les permissions dynamiques de mon application. [sur 1 point]

PAS UTILISÉ

Je sais gérer la persistance légère de mon application. [sur 1 point]

PAS UTILISÉ

Je sais gérer la persistance profonde de mon application. [sur 1 point]
Nous enregistrons le meilleur score fait par l'utilisateur grâce à de la persistance profonde. Nous utilisons plus précisément les « SharedPreferences » pour cela.
Preuve : ligne 30 de MainActivity et classe DataManager
=> 0/1_*

Je sais afficher une collection de données. [sur 1 point]

PAS UTILISÉ

Je sais coder mon propre adaptateur. [sur 2 points]

PAS UTILISÉ

Je maîtrise l'usage des fragments. [sur 2 points]

PAS UTILISÉ

Je maîtrise l'utilisation de Git. [sur 1 point]

Nous avons utilisé Git pour travailler sur ce projet en binôme. Avant chaque modification potentielle du code, il fallait donc pull le répertoire, puis à la fin de ces modifications, commit et push notre travail, pour que l'autre y ai accès à tout instant.
Preuve, les commits sur la forge, fait grâce à Git.

Application : …/20

Mon application présente un intérêt à être publié sur le store. [sur 5 points]
Notre application est DIVERTISSANTE, le système de jeu fonctionnant plutôt bien, elle pourra permettre de passer de bons moments en essayant de faire un score le plus gros possible. De plus, une partie de jeu est très rapide, on peut donc y jouer facilement dans de nombreuses situations, tout en prenant du plaisir à chaque fois, l'aspect répétitif étant inhibé par l'aspect scoring.

Mon application fonctionne de manière à être utilisée par le public. [sur 5 points]
Oui, aucun bug de fonctionnement, conflits ou autre.

Mon application utilise des contraintes spécifiées lors du choix du projet. [sur 5 points]
Nous utilisons le gyroscope du téléphone.

Mon application utilise les contraintes à bon escient. [sur 5 points]
Nous utilisons le gyroscope afin de déplacer le joueur de droite à gauche dans le jeu.